Appendix C: Categories and Examples of Risk

Plan/Schedule

Schedule is optimistic, "best case," rather than realistic, "expected case"

Plan omits necessary tasks

Schedule was based on the use of specific team members, but those team members were not available

Cannot build a product of the size specified in the time allocated

Product is larger than estimated (in lines of code, function points, or percentage of previous project’s size)

Effort is greater than estimated (per line of code, function point, module, etc.)

Re-estimation in response to schedule slips does not occur, or is overly optimistic or ignores project history

Excessive schedule pressure

A delay in one task causes cascading delays in dependent tasks

Unfamiliar or complex areas of the product take more time than expected to design and implement

Organization and Management

Project lacks an effective top-management sponsor

Layoffs and cutbacks reduce team’s capacity

Inefficient team structure reduces productivity

Lack of specific technical expertise

Management review/decision cycle is slower than expected

Budget cuts

Non-technical third-party tasks take longer than expected (control agency approvals, procurement, equipment purchase, legal reviews, etc.)

Project plans are abandoned under pressure

Inaccurate status reporting

Development Environment

Facilities are not available on time

Facilities are available but inadequate (e.g., no phones, network wiring, furniture, office supplies, etc.)

Facilities are crowded, noisy, or disruptive

Development tools are not in place by the desired time

Development tools do not work as expected; developers need time to create workarounds or to switch to new tools

Developers unfamiliar with development tools

Development tools do not provide the planned productivity

Development environment structure, policies, procedures are not clearly defined

User Involvement

User introduces new requirements after agreed upon requirements specification is complete

User finds product to be unsatisfactory

User does not buy into the project and consequently does not provide needed support

User input is not successfully solicited

User review/decision cycles for plans, prototypes, and specifications are slower than expected

User will not participate in review cycles for plans, prototypes, and specifications or is incapable of doing so

User communication time (e.g., time to answer requirements-clarification questions) is slower than expected

User-mandated support tools and environments are incompatible, have poor performance, or have inadequate functionality

User has expectations for development speed that developers cannot meet

Contractor Performance

Contractor does not deliver components when promised

Contractor delivers components of unacceptably low quality, and time must be added to improve quality

Contractor does not provide the level of domain expertise needed

Contractor does not provide the level of technical expertise needed

Requirements Management

Requirements have been base lined but continue to change

Requirements are poorly defined, and further definition expands the scope of the project

Additional requirements are added

Vaguely specified areas of the product are more time-consuming than expected

Product Characteristics

Error-prone modules require more testing, design, and implementation work than expected

Unacceptably low quality requires more testing, design, and implementation work to correct than expected

Development of flawed software functions requires redesign and implementation

Development of flawed user interface results in redesign and implementation

Development of extra software functions that are not required extends the schedule

Meeting product's size or speed constraints requires more time than expected, including time for redesign and re-implementation

Requirements for interfacing with other systems, other complex systems, or other systems that are not under the team's control result in unforeseen design, implementation, and testing

Requirement to operate under multiple operating systems takes longer to satisfy than expected

Development in an unfamiliar or unproved software environment

Development in an unfamiliar or unproved hardware environment

Dependency on a technology that is new or still under development

External Environment

Product depends on law, policy or regulations that change frequently

Multiple stakeholders outside the normal department chain of command

Key software or hardware components become unavailable, unsupported or are unexpectedly scheduled for de-support

Personnel

Acquisition of required project staff takes longer than expected

Task prerequisites (e.g., training, completion of other projects) cannot be completed on time

Poor relationships between project team and users or other stakeholders slow decision making and follow through

Lack of needed specialization (includes technical and domain knowledge) increases defects and rework

Personnel need extra time to learn unfamiliar software tools or environment

Personnel need extra time to learn unfamiliar hardware environment

Personnel need extra time to learn unfamiliar software language

Unplanned turnover of contractor key personnel

Unplanned turnover of State key personnel

New development personnel are added late in the project, and additional training and communications overhead reduces existing team members' effectiveness

Conflicts between team members

Problem team members are not removed from the team

The personnel most qualified to work on the project are not available or are not used

Personnel with critical skills needed for the project cannot be found

Key personnel are available only part time

Not enough personnel are available for the project

People's assignments do not match their strengths

Design and Implementation

Design fails to address major issues

Design requires unnecessary and unproductive implementation overhead

Flawed design

Use of unfamiliar methodology

Necessary functionality cannot be implemented using the selected methods and tools

Schedule savings from productivity enhancing tools are overestimated

Components developed separately cannot be integrated easily

Data conversion activities are underestimated or are ignored

Process

Inaccurate progress tracking

Upstream quality-assurance activities are limited or cut short

Poor quality assurance

Too little formality (lack of adherence to software policies and standards)

Too much formality (bureaucratic adherence to software policies and standards)

Weak risk management fails to detect major project risks

Project management and tracking consumes more resources than expected